

Būtent API specifikacija

Versija: refs/tags/v1.17.13

Contents

1	Būtent API veikimo principai	4
2	Autorizacija	4
2.1	Klaidų pavyzdžiai	5
3	User	5
3.1	Užklauskos	5
3.2	Struktūra	5
3.3	Pavyzdžiai	5
4	Client	5
4.1	Struktūra	6
4.1.1	Client	6
4.1.2	ClientsList	7
4.2	Klaidos	8
4.3	Pavyzdžiai	8
4.3.1	GET	8
4.3.2	POST	8
4.3.3	PUT	10
5	Good	10
5.1	Struktūra	10
5.1.1	Good [užklauskoms]	10
5.1.2	Type	11
5.1.3	Group	11
5.1.4	Good [rezultatams]	12
5.1.5	GoodsList	14
5.2	Pavyzdžiai	14
6	Prices lauko struktūra	15
6.1	Struktūra	15
6.1.1	Price	15
6.1.2	Prices	15
7	Stock	17
7.1	Struktūra	17
7.1.1	Stock užklauskoms	17
7.1.2	Stock rezultatui	18
7.1.3	StockList	18
7.2	Pavyzdžiai	18

7.2.1	Užklauso:	18
8	Discount	19
8.1	Struktūra	19
8.2	Pavyzdžiai	20
8.3	Struktūra	21
8.4	Pavyzdžiai	21
8.5	Struktūra	22
8.6	Pavyzdžiai	23
8.7	Struktūra	23
8.8	Pavyzdžiai	24
8.8.1	JSON	24
9	Trade / Turnover	24
9.1	Struktūra	25
9.1.1	Trade	25
9.1.2	Document	25
9.1.3	TradeList	26
9.2	Pavyzdžiai	26
9.2.1	GET	26
10	Trade / Sale	27
10.1	Struktūros	27
10.1.1	TradePreload	27
10.1.2	Trade	28
10.2	Item	28
10.3	Product	29
10.4	Document	29
10.4.1	CostRevenue	30
10.4.2	Transport, Duty, Other, Excise	31
10.4.3	supplierInfo, receiverInfo	31
10.5	DocumentList	32
10.6	Pavyzdžiai	32
10.6.1	GET	32
10.7	Klaidos	34
11	Trade / Payment	34
12	Trade / Payment	34
12.1	Struktūra	35
12.1.1	Payment	35
12.2	Pavyzdžiai	35
12.2.1	Užklausa:	35
12.2.2	Atsakymas:	35
13	Customer prepay	36
13.1	Struktūros	36
13.1.1	RequestSettlement	36

13.1.2	GET struktūra	36
13.1.3	POST struktūra	37
13.2	Klaidų pranešimai	38
14	Sisteminiai metodai	38
14.1	/system/version	38
14.2	/system/update	38
15	POST Rezultatas	38
15.1	Struktūra	38
15.1.1	IDs	38
15.1.2	Error	39
15.1.3	Result	39
16	Kita	40
16.1	Pavyzdžiai	40
16.1.1	Sertifikato generavimas	40

1 Būtent API veikimo principai

1. API, tai aplikacijų programavimo sąsaja (angl. Application Programming Interface, API).
2. API su “Būtent” programa komunikuoja duomenų bazės lygmenyje. Todėl turi būti užtikrinta tiesioginė **MySQL** TCP prieiga prie duomenų bazės. **MsSQL** standartinė prieiga prie duomenų bazės.
3. API gali būti įdiegtas į bet kurį serverį ar kompiuterį ir veikti kaip nepriklausomas servisas. Palaiko Linux ir Windows operacines sistemas, taip pat gali dirbti ir 32 bitų aplinkoje.
4. Užklauskos gali būti siunčiamos **JSON**, **XML**, **QUERY** ir **FORM** tipo duomenimis. Taip pat parametrus galima siųsti kartu su **Header**.
5. Atsakymas grąžinamas **JSON** ir **XML** formatais, koku formatu grąžins duomenis priklauso nuo užklauskos Headerio Content-Type. Esant Content-Type: application/xml, atsakymas bus grąžintas XML formatu, visais kitais atvejais bus grąžinamas JSON formatu.
6. Autorizacija atliekama vienu iš esamų “Būtent” programoje vartotojų. Saugumo ir atsekamumo sumetimais siūloma sukurti atskirą vartotoją su pakankamai saugiu slaptažodžiu. Šio vartotojo vardu “Būtent” programoje bus registruojami nauji įrašai.
7. API portas nustatomas konfigūracijos metu. Esant būtinybei pasiekti API iš išorės, atitinkamai turi būti atidarytas naudojamas portas.
8. **Norint importuoti dokumentus** “Būtent” programoje turi būti sukonfigūruotas “neparuošti išlaidų ir pajamų dokumentai” funkcionalumas.
9. Visos užklauskos prasideda: /api/v1/. Kur v1 nurodo naudojamą versiją. (http://ip_or_host:port/api/v1/).
10. API gali veikti HTTP ir HTTPS protokolais. Naudojant TLS turi būti sugeneruoti sertifikato raktai. Gali būti sugeneruoti Selft tipo arba vienas iš oficialių CA. Taip pat yra galimybė naudotis automatiniais sertifikatais iš LetEncrypt. Tokiu atveju, sertifikatai automatiškai bus sugeneruojami.
11. Palaiko savaiminį atsistatymą jeigu dėl kokios nors priežasties įvyko programinė klaida ir klaidos pranešimas grąžinamas su 500 klaida.
12. Palaiko HTTP Gzip suspaudimą atsakymams siųsti, galima įjungti/išjungti per konfigūraciją.
13. Įdiegtos apsaugos nuo cross-site scripting (XSS), sniffing, clickjacking, ir kitų atakų.
14. Esant dideliems užklauskų srautams, API gali išnaudoti visus maksimalius serverio ir OS resursus. Tokiu atveju siūloma API diegti atskirame serveryje, jeigu resursų neužtinka paleisti subalansuotus kelis ar keleta serverių. *Nesant techninei galimybei API gali būti apribotas procesorių kiekiu, tokiu atveju sulėtės greیتaveika.*

2 Autorizacija

Autorizacija vyksta su kiekviena nuoroda siunčiant “Būtent” programoje esančio vartotojo vardą ir slaptažodį. Autorizacija atliekama basic HTTP authentication principu.

Esant klaidingiems prisijungimo duomenims, bus grąžinama su HTTP klaida 401:

2.1 Klaidų pavyzdžiai

JSON

```
{  
  "timestamp": "2020-03-02T13:25:36.736225646+02:00",  
  "status": 401,  
  "error": "Unauthorized",  
  "path": "/api/v1/user"  
}
```

3 User

URL : /api/v1/user

Metodas : GET

Auth reikalinga : Taip

Grąžina informaciją apie vartotoją kuriuo atliekama autorizacija prie API. Visi įrašai kurie bus kuriami Būtent duomenų bazėje bus kuriami su vartotojo ID kurio prisijungimais buvo prisijungta prie API.

3.1 Užklauso

3.2 Struktūra

Lentelė: UserResponse

Nr.	Lauko pavadinimas	Tipas	Privalomas	Reikšmė
1	userID	Integer	Taip	Vartotojo kodas
2	username	String(20)	Taip	Vartotojo prisijungimo vardas

3.3 Pavyzdžiai

```
curl --request GET \  
  --url http://localhost:9666/api/v1/user \  
  --header 'authorization: Basic YXBpOnBhc3N3b3Jk'
```

```
{  
  "userID": 1,  
  "username": "api"  
}
```

4 Client

URL : /api/v1/client

Metodas : GET | POST | PUT

GET by ID : /api/v1/client/:id

Auth reikalinga : Taip

Klientų informacijos gavimui, naujų sukūrimui ir informacijos atnaujinimui. Naudojantis PUT metodu, kliento pavadinimas nebus atnaujinamas, visa kita informacija bus pakeista.

- **GET** užklausų duomenų filtravimui naudojama `Client` struktūra, rezultatas gražinamas `ClientList` struktūra.
- **POST** klientų duomenų importui naudojama `ClientList` struktūra, net jeigu bus importuojami vieno kliento duomenys.
- **PUT** kaip ir **GET**, išskyrus, privalo būti užpildytas `clientId` laukas pagal kurį ir bus atnaujinama kliento informacija.
- **DELETE** nenaudojamas, klientą galima ištrinti tik iš Būtenta programos ir tik jeigu neturi susijusių dokumentų ar kitų įrašų.

4.1 Struktūra

4.1.1 Client

Naudojama formuoti užklausoms ir rezultatams išvesti. Naudojant užklausoms reikšmės turi būti tikslios, kaip pvz: `clientId = 1, code = "123456789"` ir t.t.

GET atveju, gali būti užpildytas bet kuris vienas ir/arba keli laukai pagal kurį nurima suformuoti užklausa. Užklausoje nebūtina paduoti visus laukus, galima naudoti tik tuos kurie turie reikšmes.

POST - nenaudojamas, turi būti naudojama `ClientList` struktūra.

PUT atveju, privalomas laukas yra `clientId`, pagal jį bus atnaujinama kliento informacija duomenų bazėje. Jeigu toks kliento ID neegzistuoja arba jis nebus nurodytas, bus gražinamas klaidos pranešimas "ID not found or invalid."

Nr.	Lauko pavadinimas	Tipas	Privalomas	Reikšmė
1	<code>clientId</code>	Integer	Priklausomai nuo metodo	Kliento ID
2	<code>name</code>	String(70)	POST	Pavadinimas
3	<code>code</code>	String(30)	Ne	Kodas
4	<code>vat</code>	String(30)	Ne	PVM kodas
5	<code>bankCode</code>	String(30)	Ne	Banko kodas
6	<code>type</code>	String(30)	Ne	Kliento tipas
7	<code>discount</code>	Float(7,3)	Ne	Priskirta klientui nuolaida
8	<code>priceNumber</code>	UInt8	Ne	Kainos numerispriskirtas klientui
9	<code>index</code>	String(10)	Ne	Pašto indeksas
10	<code>address</code>	String(70)	Ne	Kliento buveinės adresas
11	<code>city</code>	String(30)	Ne	Miestas
12	<code>country</code>	String(60)	Ne	Šalis
13	<code>phone</code>	String(60)	Ne	Telefonai
14	<code>email</code>	String(100)	Ne	Elektroninis paštas
15	<code>priceSystem</code>	String(70)	Ne	Kainų sistema(Kliento pavadinimas)

clientID

GET - naudojamas jeigu norima gauti kliento informaciją pagal ID

POST - ignoruojamas ir tikrinama informacija kaip naujo kliento

PUT - privalomas

name

POST - privalomas

PUT - ingoruojamas, kliento pavadinimą galima keisti tik per Būtent programą.

JSON:

```
{
  "clientID": 1,
  "name": "Įmonės pavadinimas",
  "code": "123456780",
  "vat": "LT123456789",
  "bankCode": "40100",
  "type": "Pirkėjas",
  "discount": 10.0,
  "priceNumber": 1,
  "index": "11111",
  "address": "Gatvė Nr. 1",
  "city": "Kaunas",
  "country": "Lietuvos Respublika",
  "phone": "+37068899999",
  "email": "info@imone.lt",
  "priceSystem": "Kitos įmonės pavadinimas"
}
```

4.1.2 ClientsList

Naudojamas duomenų importui POST importuoti klientų sąrašą į duomenų bazę, taip pat naudojamas rezultatams grąžinti su metodu GET.

Nr.	Lauko pavadinimas	Tipas	Privalomas	Reikšmė
1	count	Integer	Taip	Įrašų skaičius lauke clients
2	clients	(#client-1)	Taip	Klientų sąrašas.

JSON struktūra:

```
{
  "count": 0,
  "clients": []
}
```

4.2 Klaidos

200 OK Gali būti gražinami klaidos pranešimai su struktūra Result. Galimi su klientų įkėlimu klaidų pranešimai:

- Client with name [%s] exist
- Client with code [%s] exist
- No given client name
- Unknown bank code [%s]

404 Not found Not found records. 404 Not found ID not found or invalid. 400 Bad request Not found records to import

500 Internal server error Cannot get clients, got error: %v 500 Internal server error Error on create country 500 Internal server error Error on create city %v

4.3 Pavyzdžiai

4.3.1 GET

Visų klientų sąrašo gavimas

```
curl --request GET \  
  --url http://localhost:9666/api/v1/client \  
  --header 'authorization: Basic YXBpOnBhc3N3b3Jk'
```

Kliento informacijos gavimas pagal kliento ID

```
curl --request GET --url http://localhost:9666/api/v1/client \  
  --header 'authorization: Basic YXBpOnBhc3N3b3Jk' \  
  --header 'content-type: application/json' \  
  --data '{"clientID": 1}'
```

```
curl --request GET \  
  --url http://localhost:9666/api/v1/client/1 \  
  --header 'authorization: Basic YXBpOnBhc3N3b3Jk'
```

4.3.2 POST

Užklausa:

Kelių arba vieno kliento informacijos įvedimas į Būtent programą.

```
curl --request POST \  
  --url http://localhost:9666/api/v1/client \  
  --header 'authorization: Basic YXBpOnBhc3N3b3Jk' \  
  --header 'content-type: application/json' \  
  --data '{  
    "clients": [  
      {  
        "name": "Įmonės pavadinimas",  
        "code": "123456780",  
        "vat": "LT123456789",
```



```

    "bankCode": "40100",
    "type": "Pirkėjas",
    "discount": 10.0,
    "index": "11111",
    "address": "Gatvė Nr. 1",
    "city": "Kaunas",
    "country": "Lietuvos Respublika",
    "phone": "+37068899999",
    "email": "info@imone.lt"
  }, {
    "name": "Kitas pavadinimas",
    "code": "123456789",
    "vat": "LT123456789",
    "bankCode": "40100",
    "type": "Pirkėjas",
    "discount": 10.0,
    "index": "11111",
    "address": "Gatvė Nr. 1",
    "city": "Kaunas",
    "country": "Lietuvos Respublika",
    "phone": "+37068899999",
    "email": "info@imone.lt"
  }
]
}
'

```

Atsakymas:

Atvaizduojamas atsakymas kai vienas klientas yra naujas ir kai kitas klientas jau egzistuoja. Naujam klientui suteikiamas ID ir grąžinamas kartu su kliento pavadinimu. Egzistuojantis ir/arba su klaidingais duomenimis kliento informacija grąžinama errors tag'e su klaidos pranešimu message.

```

{
  "imported": 1,
  "ids": [
    {
      "id": 1,
      "name": "Įmonės pavadinimas"
    }
  ],
  "errors": [
    {
      "message": "Client with name [Kitas pavadinimas] exist",
      "data": {
        "name": "Kitas pavadinimas",
        "code": "123456789",
        "vat": "LT123456789",

```

```

        "bankCode": "40100",
        "type": "Pirkėjas",
        "discount": 10.0,
        "index": "11111",
        "address": "Gatvė Nr. 1",
        "city": "Kaunas",
        "country": "Lietuvos Respublika",
        "phone": "+37068899999",
        "email": "info@imone.lt"
    }
}
]
}

```

4.3.3 PUT

Užklausa:

Kliento nuolaidos procento informacijos atnaujinimas. Gali būti atnaujinti visi Client struktūros tag'ai išskyrus pavadinimą.

```

curl --request PUT \
  --url http://localhost:9666/api/v1/client \
  --header 'authorization: Basic YXBpOnBhc3N3b3Jk' \
  --header 'content-type: application/json' \
  --data '{
    "clientID": 1,
    "discount": 10.0,
  }
'

```

Atsakymas:

200 OK Record updated

5 Good

URL : /api/v1/good[/:id]

Metodas : GET, POST, PUT

Auth reikalinga : Taip

Naudojamas gauti prekių nomenklatūros sąrašą

5.1 Struktūra

5.1.1 Good [užklausoms]

Naudojama užklausoms formuoti.

Nr.	Lauko pavadinimas	Tipas	Privalomas	Reikšmė
1	code	Integer	Ne	Prekės kodas
2	type	String(30)	Ne	Tipo pavadinimas
3	group	String(30)	Ne	Grupės pavadinimas
4	article	String(30)	Ne	Artikulas
5	barcode	String(30)	Ne	Bar kodas
6	image	Bool	Ne	Nuotrauka

image:

Jeigu bus nurodyta **true**, ir prie prekės bus nurodytas kelias iki nuotraukos, ji bus grąžinta užkoduota **base64**

JSON:

```
{
  "code": 0,
  "type": "",
  "group": "",
  "article": "",
  "barcode": "",
  "image": false
}
```

5.1.2 Type

Nr.	Lauko pavadinimas	Tipas	Privalomas	Reikšmė
1	id	Integer	Taip	Jrašo ID
2	name	String(100)	Taip	Pavadinimas

JSON:

```
{
  "id": 0,
  "name": ""
}
```

5.1.3 Group

Nr.	Lauko pavadinimas	Tipas	Privalomas	Reikšmė
1	id	Integer	Taip	Jrašo ID
2	name	String(100)	Taip	Pavadinimas

JSON:

```
{
  "id": 0,
  "name": ""
}
```

5.1.4 Good [rezultatams]

Nr.	Lauko pavadinimas	Tipas	Privalomas	Reikšmė
1	code	Integer	Taip (GET)	Prekės kodas
2	name	String(100)	Taip	Prekės pavadinimas
3	name_en	String(100)	Ne	Angliškas pavadinimas
4	name_ru	String(100)	Ne	Rusiškas pavadinimas
5	name_other	String(100)	Ne	Kitas pavadinimas
6	name_eshop	String(100)	Ne	Pavadinimas elektroninei prekybai/parduotuvei
7	description	String	Ne	Aprašymas
8	type	Type	Taip	Prekės tipas
9	group	Group	Taip	Prekės grupė
10	countableItem	Bool	Ne	True - prekė su skaičiuojamu likučiu
11	article	String(30)	Ne	Artikulas
12	model	String(30)	Ne	Modelis
13	certificateNo	String(30)	Ne	Sertifikato nr.
14	registrationNo	String(30)	Ne	Registr. nr.
15	temporaryNo	String(30)	Ne	Laikinas nr.
16	eShop	Bool	Ne	Elektroninės prekėspožymis
17	price	Float(15,5)	Ne	Prekės kaina
18	currency	String(5)	Ne	Valiuta
19	manufacturer	String(70)	Ne	Gamintojo pavadinimas
20	unit	String	Ne	Mato vienetas
21	gross	Float(15,5)	Ne	Bruto
22	net	Float(15,5)	Ne	Neto
23	capacity	Float(15,5)	Ne	Tūris
24	barcode	String(30)	Ne	Bar kodas
25	parentGood	Integer	Ne	Tėvinė prekė(jeigu įdiegta)
26	vat	Bool	Ne	PVM apokestinama prekė
26	image	String	Ne	Prekės paveikslėlis
27	prices	Prices	Ne	Kainos
28	custom	Struct	Ne	Nestandartiniai kliento laukų informacija
29	externalID	Interface	Ne	Išorinis ID
28	inpTime	DateTime	Ne	Įvedimo laikas

JSON:

```
{
  "code": 1,
  "name": "Prekės pavadinimas",
}
```

```
"name_en": "Good Name",
"type": {
  "id": 9,
  "name": "Tipo pavadinimas"
},
"group": {
  "id": 38,
  "name": "Grupės pavadinimas"
},
"article": "00001",
"model": "Model name",
"price": 10,
"currency": "EUR",
"manufacturer": "Gamintojo pavadinimas",
"unit": "vnt",
"gross": 2,
"net": 1.48,
"capacity": 3,
"barcode": "0705632441947",
"prices": {
  "selling": {
    "amount": 10,
    "currency": "EUR"
  },
  "prime": {
    "amount": 0.151,
    "currency": "EUR"
  },
  "price1": {
    "amount": 1,
    "currency": "EUR"
  },
  "price2": {
    "amount": 2,
    "currency": "EUR"
  },
  "price3": {
    "amount": 3,
    "currency": "EUR"
  },
  "price4": {
    "amount": 4,
    "currency": "EUR"
  },
  "price5": {
    "amount": 5,
    "currency": "EUR"
  }
}
```

```

    },
    "price6": {
      "ammount": 6,
      "currency": "EUR"
    },
    "price7": {
      "ammount": 7,
      "currency": "EUR"
    },
    "price8": {
      "ammount": 8,
      "currency": "EUR"
    },
    "price9": {
      "ammount": 9,
      "currency": "EUR"
    },
    "price10": {
      "ammount": 10,
      "currency": "EUR"
    }
  },
  "inpTime": "2014-05-20 11:23:54"
}

```

5.1.5 GoodsList

Naudojamas grąžinti prekių nomenklatūros rezultatui.

Nr.	Lauko pavadinimas	Tipas	Privalomas	Reikšmė
1	count	Integer	Taip	Jrašų skaičius
2	goods	(#good-1)	Taip	Prekių sąrašas

```

{
  "count": 0,
  "goods": []
}

```

5.2 Pavyzdžiai

Užklausa:

GET

Norint gauti visą sąrašą

```
curl --request GET \
  --url http://localhost:9666/api/v1/good \
  --header 'authorization: Basic YXBpOnBhc3N3b3Jk'
```

Norint ištraukti prekės informaciją kurios kodas yra 1 (Būtent programoje Prekė=1)

```
curl --request GET \
  --url http://localhost:9666/api/v1/good \
  --header 'authorization: Basic YXBpOnBhc3N3b3Jk' \
  --header 'content-type: application/json' \
  --data '{"code": 1}'
```

```
curl --request GET \
  --url http://localhost:9666/api/v1/good/1 \
  --header 'authorization: Basic YXBpOnBhc3N3b3Jk'
```

Filtruoti duomenis pagal parametrus

```
curl --request GET \
  --url http://localhost:9666/api/v1/good \
  --header 'authorization: Basic YXBpOnBhc3N3b3Jk' \
  --header 'content-type: application/json' \
  --data '{
    "eshop": true,
    "type": "Tipo pavadinimas",
    "group": "Grupės pavadinimas"
  }'
```

6 Prices lauko struktūra

6.1 Struktūra

6.1.1 Price

Nr.	Lauko pavadinimas	Tipas	Reikšmė
1	ammount	Float	Kaina
2	currency	String	Valiuta

6.1.2 Prices

Kokia kainos informacija yra nuo 1 iki 10 priklauso nuo kliento programos konfigūracijos.

Nr.	Lauko pavadinimas	Tipas	Privalomas	Reikšmė
1	selling	Price	Ne	Pardavimo kaina
2	prime	Price	Ne	Savikaina
3	price1	Price	Ne	Kaina 1
4	price2	Price	Ne	Kaina 2
5	price3	Price	Ne	Kaina 3

Nr.	Lauko pavadinimas	Tipas	Privalomas	Reikšmė
6	price4	Price	Ne	Kaina 4
7	price5	Price	Ne	Kaina 5
8	price6	Price	Ne	Kaina 6
9	price7	Price	Ne	Kaina 7
10	price8	Price	Ne	Kaina 8
11	price9	Price	Ne	Kaina 9
12	price10	Price	Ne	Kaina 10

JSON

```
{
  "selling": {
    "ammount": 10,
    "currency": "EUR"
  },
  "prime": {
    "ammount": 0.151,
    "currency": "EUR"
  },
  "price1": {
    "ammount": 1,
    "currency": "EUR"
  },
  "price2": {
    "ammount": 2,
    "currency": "EUR"
  },
  "price3": {
    "ammount": 3,
    "currency": "EUR"
  },
  "price4": {
    "ammount": 4,
    "currency": "EUR"
  },
  "price5": {
    "ammount": 5,
    "currency": "EUR"
  },
  "price6": {
    "ammount": 6,
    "currency": "EUR"
  },
  "price7": {
    "ammount": 7,
```



```

    "currency": "EUR"
  },
  "price8": {
    "ammount": 8,
    "currency": "EUR"
  },
  "price9": {
    "ammount": 9,
    "currency": "EUR"
  },
  "price10": {
    "ammount": 10,
    "currency": "EUR"
  }
}

```

7 Stock

URL: /api/v1/stock[/:good[/:store]]

Metodas: GET

Auth reikalinga: Taip

Naudojama prekių likučiams gauti. Grąžina tik teigiamus [kiekis > 0] likučius.

7.1 Struktūra

7.1.1 Stock užklausoms

Naudojama filtruoti likučių duomenis.

Nr.	Lauko pavadinimas	Tipas	Privalomas	Reikšmė
1	good	Integer	Ne	Prekės kodas
2	goods	Array Integer	Ne	Prekių kodai
3	mode	String(10)	Ne	Sandėlio grupės pagal simbolį
4	store	String(10)	Ne	Sandėlio kodas
5	reserved	Bool	Ne	Rezervuotos prekės
6	zero_qty	Bool	Ne	Nuliniai kiekiai

```

{
  "good": 0,
  "goods": [1, 2, 3],
  "mode": "",
  "store": "",
  "reserved": true,
  "zero_qty": true
}

```

7.1.2 Stock rezultatui

Naudojama struktūra prekių likučių grąžinimui struktūroje StockList.

Nr.	Lauko pavadinimas	Tipas	Privalomas	Reikšmė
1	good	Integer	Taip	Prekės kodas
2	store	String(10)	Taip	Sandėlio kodas
3	quantity	Float(15,5)	Taip	Kiekis
4	reserved	Float(15,5)	Ne	Rezervuotas kiekis
5	unit	String	Taip	Mato vienetas

JSON

```
{
  "good": 1,
  "store": "S1",
  "quantity": 10.111
}
```

7.1.3 StockList

Grąžinamas prekių likučių rezultatas.

Nr.	Lauko pavadinimas	Tipas	Privalomas	Reikšmė
1	count	Integer	Taip	Įrašų skaičius
2	stocks	[]Stock	Taip	Likučių sąrašas

```
{
  "count": 0,
  "stocks": [
    /* ... */
  ]
}
```

7.2 Pavyzdžiai

7.2.1 Užklauso:

Visiems likučiams gauti

```
curl --request GET \
  --url http://localhost:9666/api/v1/stock \
  --header 'authorization: Basic YXBpOnBhc3N3b3Jk' \
  --header 'content-type: application/json'
```

Gauti likučius ir rezervuotus likučių kiekius. Rezervuotus kiekis reikia minusuoti iš likučio.

```
curl --request GET \  
  --url http://localhost:9666/api/v1/stock \  
  --header 'authorization: Basic YXBpOnBhc3N3b3Jk' \  
  --header 'content-type: application/json' \  
  --data '{  
    "reserved": true  
  }'
```

Konkrečios prekės likučiui gauti

```
curl --request GET \  
  --url http://localhost:9666/api/v1/stock \  
  --header 'authorization: Basic YXBpOnBhc3N3b3Jk' \  
  --header 'content-type: application/json' \  
  --data '{  
    "good": 1  
  }'
```

Kelių prekių likučiams gauti

```
curl --request GET \  
  --url http://localhost:9666/api/v1/stock \  
  --header 'authorization: Basic YXBpOnBhc3N3b3Jk' \  
  --header 'content-type: application/json' \  
  --data '{  
    "goods": [1, 2, 3]  
  }'
```

Kelių/keletos sandėlių likučių gavimui. Bus ištraukti likučiai kur sandėlys turės požymį "A".

```
curl --request GET \  
  --url http://localhost:9666/api/v1/stock \  
  --header 'authorization: Basic YXBpOnBhc3N3b3Jk' \  
  --header 'content-type: application/json' \  
  --data '{  
    "mode": "A"  
  }'
```

8 Discount

URL : /api/v1/discount/client[/:id]

Metodas : GET

Auth reikalinga : Taip

Grąžina informaciją apie individualias kainas prekėms, nuolaidos procentą prekių grupei arba tipui.

8.1 Struktūra

Lentelė: Discount [užklausoms, rezultatui]

Nr.	Lauko pavadinimas	Tipas	Privalomas	Reikšmė
1	id	Integer	taip	Nuolaidos kodas
2	client	String(70)	taip	Klientas
3	clientID	UInteger	taip	Kliento ID
4	dateFrom	DateTime	ne	Data nuo
5	dateTo	DateTime	ne	Data iki
6	type	String(30)	ne	Prekės tipas
7	typeID	UInteger	ne	Prekės tipo ID
8	group	String(60)	ne	Prekės grupė
9	groupID	UInteger	ne	Prekės grupės ID
10	pricePosition	String(20)	ne	Kainos pozicija
11	model	String(30)	ne	Modelis
12	supplier	String(70)	ne	Tiekėjas
13	manufacturer	String(70)	ne	Gamintojas
14	goodCode	Integer	ne	Prekės kodas
15	article	String(30)	ne	Prekės artikulius
16	barcode	String(30)	ne	Barkodas
17	quantityFrom	Float(15,5)	ne	Kiekis nuo
18	quantityTo	Float(15,5)	ne	Kiekis iki
19	percent	Float(8,3)	ne	Procentas
20	discountAmount	Float(15,5)	ne	Suma
21	price	Float(15,5)	ne	Kaina
22	currency	String(5)	ne	Valiuta
23	clientPrice	Float(12,2)	ne	Kliento kaina
24	clientCurrency	String(5)	ne	Kliento Valiuta
25	currencyPrice	Float(12,2)	ne	Kaina Valiuta
26	currencyCurrency	String(5)	ne	Valiutos valiuta
27	discountMode	String(10)	ne	Nuolaidos būdas

8.2 Pavyzdžiai

```
curl --request GET \
  --url http://localhost:9666/api/v1/discount \
  --header 'authorization: Basic YXBpOnBhc3N3b3Jk'

{
  "count": 1,
  "discounts": [
    {
      "id": 1,
      "client": "Įmonės pavadinimas",
      "type": "Nuolaidos tipas",
      "percent": 5
    }
  ]
}
```

URL : /api/v1/discount/good[:id]

Metodas : GET

Auth reikalinga : Taip

Grąžina informaciją apie nuolaidas, kainas prekėms.

8.3 Struktūra

Lentelė: goodDiscount [užklausoms, rezultatui]

Nr.	Lauko pavadinimas	Tipas	Privalomas	Reikšmė
1	id	Integer	taip	Nuolaidos kodas
2	goodCode	Integer	taip	Prekė
3	client	String(70)	ne	Kliento pavadinimas
4	clientID	UInteger	ne	Kliento ID
5	dateFrom	DateTime	ne	Data nuo
6	dateTo	DateTime	ne	Data iki
7	clientType	String(30)	ne	Klientų tipas
8	clientGroup	String(30)	ne	Klientų grupė
9	clientMode	String(10)	ne	Klientų savybės
10	warehouse	String(5)	ne	Sandėlis
11	warehouseType	String(5)	ne	Sandėlių tipas
12	warehouseGroup	String(5)	ne	Sandėlių grupė
13	warehouseMode	String(10)	ne	Sandėlių savybės
14	letter	String(1)	ne	Raidė
15	operation	String(60)	ne	Operacija
16	operationMode	String(10)	ne	Operacijos būdas
17	paymentType	String(30)	ne	Atsiskaitymas
18	quantityFrom	Float(15,5)	ne	Kiekis nuo
19	quantityTo	Float(15,5)	ne	Kiekis iki
20	unit	String(10)	ne	Mato vien.
21	priceNo	[U]Integer	ne	Kainos numeris
22	discountPercent	Float(7,3)	ne	Nuolaidos procentas
23	discountAmount	Float(15,5)	ne	nuolaidos suma
24	price	Float(15,5)	ne	Kaina
25	currency	String(5)	ne	Valiuta
26	discountMode	String(10)	ne	Nuolaidos būdas

8.4 Pavyzdžiai

```
curl --request GET \
  --url http://localhost:9666/api/v1/discount/good \
  --header 'authorization: Basic YXBpOnBhc3N3b3Jk'

{
  "count": 1,
```

```

"goodDiscounts": [
  {
    "id": 1,
    "goodCode": 999,
    "clientType": "kliento tipas",
    "discountPercent": 20
  }
]
}

```

URL : /api/v1/discount/group[/:id]

Metodas : GET

Auth reikalinga : Taip

Grąžina informaciją apie nuolaidas prekių grupėms.

8.5 Struktūra

Lentelė: groupDiscount [užklausoms, rezultatui]

Nr.	Lauko pavadinimas	Tipas	Privalomas	Reikšmė
1	id	Integer	taip	Nuolaidos kodas
2	group	String(60)	ne	Prekės grupė
3	client	String(70)	ne	Klientas
4	clientID	UInteger	ne	Kliento ID
5	dateFrom	DateTime	ne	Data nuo
6	dateTo	DateTime	ne	Data iki
7	article	String(30)	ne	Prekės artikuląs
8	clientType	String(30)	ne	Klientų tipas
9	clientGroup	String(30)	ne	Klientų grupė
10	clientMode	String(10)	ne	Klientų savybės
11	warehouse	String(5)	ne	Sandėlis
12	warehouseType	String(5)	ne	Sandėlių tipas
13	warehouseGroup	String(5)	ne	Sandėlių grupė
14	warehouseMode	String(10)	ne	Sandėlių savybės
15	letter	String(1)	ne	Raidė
16	operation	String(60)	ne	Operacija
17	operationMode	String(10)	ne	Operacijos būdas
18	paymentType	String(30)	ne	Atsiskaitymas
19	quantityFrom	Float(15,5)	ne	Kiekis nuo
20	quantityTo	Float(15,5)	ne	Kiekis iki
21	unit	String(10)	ne	Mato vien.
22	priceNo	Integer(2)	ne	Kainos numeris
23	discountPercent	Float(7,3)	ne	Nuolaidos procentas
24	discountAmount	Float(15,5)	ne	nuolaidos suma
25	currency	String(5)	ne	Valiuta

Nr.	Lauko pavadinimas	Tipas	Privalomas	Reikšmė
26	discountMode	String(10)	ne	Nuolaidos būdas

8.6 Pavyzdžiai

```
curl --request GET \
  --url http://localhost:9666/api/v1/discount/group \
  --header 'authorization: Basic YXBpOnBhc3N3b3Jk'
```

```
{
  "count": 1,
  "discounts": [
    {
      "id": 1,
      "group": "grupes pavadinimas",
      "clientType": "kliento tipas",
      "discountPercent": 10
    }
  ]
}
```

URL: /api/v1/discount/type[/:id]

Metodas: GET

Auth reikalinga: Taip

Grąžina informaciją apie nuolaidas prekių tipams.

8.7 Struktūra

Lentelė: typeDiscount [užklausoms, rezultatui]

Nr.	Lauko pavadinimas	Tipas	Privalomas	Reikšmė
1	id	Integer	taip	Nuolaidos kodas
2	type	String(30)	ne	Prekės tipas
3	client	String(70)	ne	Klientas
3	clientID	UInteger	ne	Kliento ID
4	dateFrom	DateTime	ne	Data nuo
5	dateTo	DateTime	ne	Data iki
6	article	String(30)	ne	Prekės artikulius
7	clientType	String(30)	ne	Klientų tipas
8	clientGroup	String(30)	ne	Klientų grupė
9	clientMode	String(10)	ne	Klientų savybės
10	warehouse	String(5)	ne	Sandėlis
11	warehouseType	String(5)	ne	Sandėlių tipas
12	warehouseGroup	String(5)	ne	Sandėlių grupė
13	warehouseMode	String(10)	ne	Sandėlių savybės

Nr.	Lauko pavadinimas	Tipas	Privalomas	Reikšmė
14	letter	String(1)	ne	Raidė
15	operation	String(60)	ne	Operacija
16	operationMode	String(10)	ne	Operacijos būdas
17	paymentType	String(30)	ne	Atsiskaitymas
18	quantityFrom	Float(15,5)	ne	Kiekis nuo
19	quantityTo	Float(15,5)	ne	Kiekis iki
20	mato_vien	String(10)	ne	Mato vien.
21	priceNo	Integer(2)	ne	Kainos numeris
22	discountPercent	Float(7,3)	ne	Nuolaidos procentas
23	discountAmount	Float(15,5)	ne	nuolaidos suma
24	currency	String(5)	ne	Valiuta
25	discountMode	String(10)	ne	Nuolaidos būdas

8.8 Pavyzdžiai

8.8.1 JSON

```
curl --request GET \
  --url http://localhost:9666/api/v1/discount/type \
  --header 'authorization: Basic YXBpOnBhc3N3b3Jk'

{
  "typeDiscounts": [
    {
      "id": 1,
      "type": "Nuolaidos tipas",
      "client": "Įmonės pavadinimas",
      "article": "XXXXX"
    }
  ]
}
```

9 Trade / Turnover

URL : /api/v1/trade/turnover

Metodas : GET

Auth reikalinga : Taip

Metodas yra skirtas gauti nurodyto kliento apyvartai nurodytu laikotarpiu. Kreipinyje privalomai turi būti užpildytas vienas iš šių parametrų: dateFrom, dateTo, client. Kai [debt = True], bus grąžinti tik tie dokumentai kurie yra neapmokėti.

9.1 Struktūra

9.1.1 Trade

Užklausoms formuoti.

Nr.	Lauko pavadinimas	Tipas	Privalomas	Reikšmė
1	dateFrom	String(10)	*Ne	Data nuo
2	dateTo	String(10)	*Ne	Data iki
3	recipient	String(70)	*Ne	Gavėjas
4	debt	Bool	Ne	Kai True , bus gražinti tik tie dokumentai kur skola > 0 .

***Ne** - Vienas iš laukų *dateFrom*, *dateTo* arba *receiver* privalo būti užpildytas, kitu atveju bus gražinamas klaidos pranešimas ir užklausa nevykdoma.

```
{
  "dateFrom": "",
  "dateTo": "",
  "recipient": "",
  "debt": false
}
```

9.1.2 Document

Pardavimo dokumentas

Nr.	Lauko pavadinimas	Tipas	Privalomas	Reikšmė
1	date	DateTime	Taip	Dokumento data
2	documentNo	String	Taip	Dokumento Nr.
3	series	String	Ne	Serija
4	otherDoc	String	Ne	Kitas dokumento Nr.
5	recipient	String(70)	Taip	Gavėjo pavadinimas
6	manager	Integer	Ne	Vadybininko tabelio Nr.
7	total	Float	Taip	Dokumento suma
8	currency	String	Taip	Valiuta
9	terms	String	Ne	Apmokėjimo terminas
10	paymentAmount	Float	Ne	Apmokėta suma
11	paymentCurrency	String	Ne	Apmokėjimo valiuta
12	paymentDate	DateTime	Ne	Apmokėjimo data

```
{
  "date": "",
  "documentNo": "",
  "series": "",
  "otherDoc": "",
  "recipient": "",
  "manager": 0,
  "total": 100.00,
}
```

```

    "currency": "EUR",
    "terms": "",
    "paymentAmount": 100.00,
    "paymentCurrency": "EUR",
    "paymentDate": "",
    "debt": 50.00
}

```

9.1.3 TradeList

Pardavimo dokumentų sąrašo gražinimui.

Nr.	Lauko pavadinimas	Tipas	Privalomas	Reikšmė
1	count	Integer	Taip	Jrašų skaičius
2	documents	[]Document	Taip	Dokumentų sąrašas

```

{
  "count": 0,
  "documents": [
    /* ... */
  ]
}

```

9.2 Pavyzdžiai

9.2.1 GET

Užklausa:

```

curl --request GET \
  --url http://localhost:9666/api/v1/trade/turnover \
  --header 'authorization: Basic YXBpOnBhc3N3b3Jk' \
  --header 'content-type: application/json' \
  --data '{
    "dateFrom": "2016-01-01",
    "debt": true
  }'

```

Atsakymas:

```

{
  "count": 1,
  "documents": [
    {
      "date": "",
      "documentNo": "",
      "series": "",
      "otherDoc": "",
    }
  ]
}

```

```

    "recipient": "",
    "manager": 0,
    "total": 100.00,
    "currency": "EUR",
    "terms": "",
    "paymentAmount": 100.00,
    "paymentCurrency": "EUR",
    "paymentDate": "",
    "debt": 50.00
  }
]
}

```

10 Trade / Sale

GET : /api/v1/trade/[sale/presale/purchase/repurchase[/:id]]

POST : /api/v1/trade/[sale/presale/purchase/repurchase]

Metodas : GET | POST

Auth reikalinga : Taip

Pardavimo dokumentų gavimui, dokumentai gali būti grąžinami su kliento informacija, prekėmis ir apmokėjimais.

sale - Jau įrašyti dokumentai kurie turi įtakos prekių likučiams ir skoloms

presale - Neparuošti išlaidų dokumentai, likučių neįtakoja, stock metode atvaizduojami lauke reserved

repurchase - Neparuošti pajamavimo dokumentai, likučiams įtakos neturi.

10.1 Struktūros

10.1.1 TradePreload

Papildomų duomenų grąžinimui su Trade rezultatais. Naudojamas Trade struktūroje.

Nr.	Lauko pavadinimas	Tipas	Privalomas	Reikšmė
1	items	Bool	Ne	Grąžina dokumento prekes
2	payments	Bool	Ne	Grąžina visus dokumente esančius apmokėjimus
3	suppliers	Bool	Ne	Kliento informacija pagal tiekėjus
4	recipients	Bool	Ne	Kliento informacija pagal gavėjus

```

{
  "items": false,
  "payments": false,
  "suppliers": false,

```

```
"recipients": false
}
```

10.1.2 Trade

Užklausoms formuoti.

Lauko pavadinimas	Tipas	Privalomas	Reikšmė
documentID	Integer	Ne	Būtent programos dokumento ID
externalID	Integer	Ne	Filtravimas pagal externalID
dateFrom	String(10)	*Ne	Data nuo
dateTo	String(10)	*Ne	Data iki
supplier	String(70)	Ne	Tiekėjas
recipient	String(70)	*Ne	Gavėjas
debt	Bool	Ne	Kai True , bus grąžinti tik tie dokumentai kur skola > 0 .
payment	Bool	Ne	Tik apmokėjimą turintiems dokumentą
preload	TradePreload	Ne	Papildomų duomenų užkrovimui

**Ne - Vienas iš laukų dateFrom, dateTo arba receiver privalo būti užpildytas, kitu atveju bus grąžinamas klaidos pranešimas ir užklausa nevykdoma.*

```
{
  "dateFrom": "2016-01-02",
  "dateTo": "2016-01-02",
  "supplier": null,
  "receiver": null,
  "debt": false,
  "payments": false,
  "preload": {
    "items": false,
    "payments": false,
    "suppliers": false,
    "receivers": false
  }
}
```

10.2 Item

Lauko pavadinimas	Tipas	Privalomas	Reikšmė
code	Integer	Taip	Prekės kodas
article	String	Ne	Artikulas
article_2	String	Ne	Artikulas 2
volume	Float	Ne	Tūris
density	Float	Ne	Tankis
quantity	Float	Taip	Kiekis
unit	String	Taip	Mato vieneto pavadinimas

Lauko pavadinimas	Tipas	Privalomas	Reikšmė
price	Float	Taip	Kaina
currency	String	Taip	Valiuta
discount	Float	Ne	Nuolaida procentais arba suma
discMode	String	Ne	* - procentais, kai nenurodyta - suma
certificat	String	Ne	Sertifikato Nr.
exciseQuantity	Float	Ne	Akcizo kiekis
exciseTotal	Float	Ne	Akcizo suma
custom	Struct	Ne	Nestandardiniai kliento laukai
vat	Vat	Ne	PVM struktūra

10.3 Product

Lauko pavadinimas	Tipas	Privalomas	Reikšmė
code	Integer	Taip	Prekės kodas
article	String	Ne	Artikulas
article_2	String	Ne	Artikulas 2
quantity	Float	Taip	Kiekis
unit	String	Taip	Mato vieneto pavadinimas
price	Float	Ne	Kaina
currency	String	Ne	Valiuta
discount	Float	Ne	Nuolaida procentais arba suma
discMode	String	Ne	* - procentais, kai nenurodyta - suma
vat	Vat	Ne	PVM struktūra
primeCost	Float	Ne	Savikaina
primeCurrency	String	Ne	Savikainos valiuta
primeCost2	Float	Ne	Savikaina 2
primeCurrency2	String	Ne	Savikainos 2 valiuta
notes	String	Ne	Pastabos

10.4 Document

Dokumento informacija, naudojama struktūroje DocumentList.

Lauko pavadinimas	Tipas	Privalomas	Reikšmė
documentID	Integer	Taip	Dokumento ID
externalID	Integer	Ne	Išorinis ID
letter	String	Ne	Dokumento raidė
date	DateTime	Taip	Dokumento data
no	String	Taip	Dokumento Nr.
series	String	Ne	Dokumento serija
otherDoc	String	Ne	Kitas dokumento numeris
otherDoc2	String	Ne	Kitas dokumento numeris 2
operation	String	Taip	Operacija

Lauko pavadinimas	Tipas	Privalomas	Reikšmė
supplierName	String	Ne	Tiekėjo pavadinimas
supplierInfo	Client	Ne	Kliento informacija pagal tiekėjo pavadinimą
receiverName	String	Ne	Gavėjo pavadinimas
receiverInfo	Client	Ne	Kliento informacija pagal gavėjo pavadinimą
store	String	Taip	Iš kokio sandėlio
loadingTime	DateTime	Ne	Pakrovimo data ir laikas
loadingPlace	String	Ne	Pakrovimo vieta
destinationTime	DateTime	Ne	Pristatymo data ir laikas
destinationPlace	String	Ne	Pristatymo vieta
employeeCode	Integer	Ne	Darbuotojo tabelio Nr.
managerCode	Integer	Ne	Vadybininko tabelio Nr.
car	String	Ne	Automobilis
driver	String	Ne	Vairuotojas
transportClient	String	Ne	Pervežimo įmonė
total	Float	Taip	Dokumento suma
currency	String	Taip	Valiuta
subTotal	Float	Ne	Suma be PVM
vatTotal	Float	Ne	PVM suma
terms	DateTime	Ne	Apmokėjimo terminas
paymentAmount	Float	Ne	Apmokėta suma
paymentCurrency	String	Ne	Apmokėjimo valiuta
paymentDate	DateTime	Ne	Apmokėjimo data
debt	Float	Ne	Kliento skola už dokumentą
items	[]Item	Ne	Parduotų prekių sąrašas
products	[]Product	Ne	Gaminių sąrašas
payments	[]Payment	Ne	Mokėjimo dokumentų sąrašas
exchangeRate	Float	Ne	Valiutos kursas
exchangeCurrency	String	Ne	Valiutos kurso valiuta
costRevenue	CostRevenue	Ne	Pajamavimo sąnaudos

POST atveju `series` laukas tikrinimas jeigu sukonfigūruotas tikrinimas konfigūracijoje. Kai `series` nurodyta reikšmė ir `otherDoc` tuščia reikšmė, automatiškai sugeneruojama `otherDoc` reikšmė sekančiu iš eilės einančiu numeriu. Numerio ilgis formuojamas pagal serijos lentelėje nurodytą ilgį.

10.4.1 CostRevenue

Lauko pavadinimas	Tipas	Privalomas	Reikšmė
transport	[Transport]	Ne	Transportavimo sąnaudos
duty	[Duty]	Ne	Muito sąnaudos
other	[Other]	Ne	Kitos sąnaudos
excise	[Excise]	Ne	Akcizas
primaryCurrency	String	Ne	Savikainos valiuta
rateAmount	Float	Ne	Valiutos kursas

Lauko pavadinimas	Tipas	Privalomas	Reikšmė
rateCurrency	String	Ne	Kurso valiuta

10.4.2 Transport, Duty, Other, Excise

Lauko pavadinimas	Tipas	Privalomas	Reikšmė
enable	Bool/null	Ne	Įtraukti į savikainos skaičiavimą
date	DateTime	Ne	Data
amount	Float	Ne	Suma
currency	String	Ne	Valiuta
clientName	String	Ne	Kliento pavadinimas
clientInfo	Client	Ne	Kliento kortelė
debit	String	Ne	Debetinė sąskaita
credit	String	Ne	Kreditinė sąskaita

10.4.3 supplierInfo, receiverInfo

GET - Kliento informacija pagal kliento pavadinimą.

POST - Jeigu informacija užpildyta, ignoruojamas laukas supplierName ir jeigu klientas neegzistuoja duomenų bazėje, sukuriamas naujas, jeigu egzistuoja priskiriamas pavadinimas laukui supplierName.

Papildoma konfigūracija kaip turėtų veigti jungtis. Pagal nutylėjimą, visos reikšmės yra **False**

document:

```

saleSeries: true      \\ Pardavimo dokumentams
preSaleSeries: true  \\ Neparuoštiems pardavimo dokumentams
saleToPreSale: true  \\ Jeigu trūksta likučio dokumentas perkeliamas į neparuoštus

```

```

{
  "documentID": 1,
  "externalID": 1,
  "letter": "A",
  "date": "2019-03-21 00:00:00",
  "no": "11111111",
  "operation": "Pardavimas",
  "supplierName": "Tiekėjo pavadinimas",
  "supplierInfo": null,
  "receiverName": "Gavėjo pavadinimas",
  "receiverInfo": null,
  "store": "S1",
  "loadingPlace": "Adresas, vieta",
  "destinationTime": "2019-03-21 00:00:00",
  "destinationPlace": "Pristatymo adresas",
  "employeeCode": 1,
  "managerCode": 1,

```

```

"car": "AAA 000",
"driver": "Vairuotojas",
"transportClient": "Transporto įmonė",
"total": 377.56,
"currency": "EUR",
"terms": "2019-04-11T00:00:00+03:00",
"paymentAmount": 16.83,
"paymentCurrency": "EUR",
"paymentDate": "2019-03-21T00:00:00+02:00",
"debt": 360.73,
"items": null,
"payments": null,
}

```

10.5 DocumentList

Dokumentų sąrašas

Nr.	Lauko pavadinimas	Tipas	Privalomas	Reikšmė
1	count	Integer	Taip	Įrašų skaičius
2	documents	[]Document	Taip	Dokumentų sąrašas

```

{
  "count": 0,
  "documents": [
    /* ... */
  ]
}

```

10.6 Pavyzdžiai

10.6.1 GET

Užklausa

```

curl --request GET \
  --url http://localhost:9666/api/v1/trade/sale \
  --header 'authorization: Basic YXBpOnBhc3N3b3Jk' \
  --header 'content-type: application/json' \
  --data '{
    "dateFrom": "2019-03-21",
    "payment": true,
    "preload": {
      "items": true
    }
  }'

```


Atsakymas

```
{
  "count": 3,
  "documents": [
    {
      "documentID": 11853074,
      "externalID": 192466537,
      "letter": "2",
      "date": "2019-03-21 00:00:00",
      "no": "11111111",
      "operation": "Pardavimas",
      "receiverName": "Gavėjas",
      "store": "S1",
      "loadingPlace": "Gatvė 1",
      "destinationTime": "2019-03-21 00:00:00",
      "destinationPlace": "Miestas",
      "employeeCode": 1,
      "managerCode": 2,
      "car": "Future car, AAA 111",
      "driver": "Vairuotojas",
      "transportClient": "Logistikos įmonė",
      "total": 377.56,
      "currency": "EUR",
      "terms": "2019-04-11T00:00:00+03:00",
      "paymentAmount": 16.83,
      "paymentCurrency": "EUR",
      "paymentDate": "2019-03-21T00:00:00+02:00",
      "debt": 360.73,
      "items": [
        {
          "code": 1111,
          "article": "prekė",
          "quantity": 20,
          "unit": "vnt",
          "price": 1.61,
          "currency": "EUR",
          "vat": {
            "status": "S",
            "value": 21,
            "mode": "%"
          }
        }
      ]
    }
  ]
}
```

10.7 Klaidos

200 OK Gali būti grąžinami klaidos pranešimai su struktūra Result. Galimi įkėlimo klaidų pranešimai:

- Document date is required
- Wrong document date [%v]
- Unknown store [%v]
- Unknown manager ID [%v]
- Unknown driver ID [%v]
- Not found items
- Empty external ID
- Document with ID [%v] already exists
- Unknown measure unit [%v]
- Unknown measure for Item [%v]
- Unknown currency [%v]
- Unknown Item ID [%v]
- It is not sale operation
- It is not purchase operation

11 Trade / Payment

GET : /api/v1/trade/:documentID

Metodas : GET

Auth reikalinga : Taip

Grąžina bet kokį dokumentą iš apyvartos pagal dokumento ID. *Užklauso ir struktūra atitinka PreSale ir PrePurchase.*

12 Trade / Payment

GET : /api/v1/trade/payment/:id

GET : /api/v1/trade/payment/document/:documentID

POST : /api/v1/trade/payment

PUT : /api/v1/trade/payment

Metodas : GET | POST | PUT

Auth reikalinga : Taip

Apmokėjimų gavimui, importui, koregavimui. Apmokėjimas gali būti įkeltas visiems dokumentų tipams (pirkimams, pardavimas, neparuoštiems dokumentams). Privalo būti nurodytas dokumentID pagal kurį ir bus priskirtas apmokėjimas. Dokumento suma ir galutinė apmokėjimo suma nėra tikrinamos, todėl gali būti ir permoka.

GET grąžinami dokumentai pagal paymentID arba documentID, pastaruoju atveju bus grąžinti visi dokumento apmokėjimai

POST privalo būti užpildytas `documentID`, jeigu nebus užpildytas šis laukas bus grąžinamas klaidos pranešimas prie įrašo.

PUT privalo būti užpildytas `paymentID`, jeigu nėra užpildytas bus grąžinamas klaidos pranešimas.

Įterpinėjant ir/arba koreguojant įrašą laukas `account` arba laukas `paymentMethod` turi būti užpildytas.

12.1 Struktūra

12.1.1 Payment

Papildomų duomenų grąžinimui su Trade rezultatais. Naudojamas Trade struktūroje.

Nr.	Lauko pavadinimas	Tipas	Privalomas	Reikšmė
1	<code>paymentID</code>	Integer	Kintamas	Apmokėjimo ID
2	<code>documentID</code>	Integer	Taip	Dokumento ID
3	<code>document</code>	String	Ne	Dokumento numeris
4	<code>date</code>	DateTime	Taip	Apmokėjimo data
5	<code>client</code>	String	Ne	Mokėtojo pavadinimas iš žinyno
6	<code>amount</code>	Float	Taip	Mokėjimo suma
7	<code>currency</code>	String	Taip	Apmokėjimo valiuta
8	<code>account</code>	String	Ne	Buhalterinės sąskaitos numeris
9	<code>content</code>	String	Ne	Turinys
10	<code>paymentMethod</code>	String	Ne	Apmokėjimo būdas iš žinyno
11	<code>error</code>	String	Ne	Klaidos pranešimas

12.2 Pavyzdžiai

12.2.1 Užklausa:

```
curl --request GET \
  --url http://localhost:9666/api/v1/trade/payment/78467 \
  --header 'authorization: Basic YXBpOnBhc3N3b3Jk'
```

12.2.2 Atsakymas:

```
{
  "count": 1,
  "payments": [
    {
      "paymentID": 78467,
      "documentID": 104256,
      "document": null,
      "date": "2020-04-27 00:00:00",
      "amount": 4305.35,
      "currency": "EUR",
      "account": "27111"
    }
  ]
}
```

```

    }
  ]
}

```

13 Customer prepay

GET : /api/v1/settlement/customer/prepay[/:id]

POST : /api/v1/settlement/customer/prepay

Auth reikalinga : Taip

Pirkėjų avansai

13.1 Struktūros

13.1.1 RequestSettlement

Užklausoms formuoti.

Nr.	Lauko pavadinimas	Tipas	Privalomas	Reikšmė
1	id	Integer	Ne	Būtent programos dokumento ID
1	dateFrom	String(10)	Ne	Data nuo
2	dateTo	String(10)	Ne	Data iki
3	client	String(70)	Ne	Kliento pavadinimas

id - Jeigu yra nurodomas ID tag'as, kiti tag'ai yra ignoruojami

```

{
  "dateFrom": "2021-09-07",
  "dateTo": "2021-09-09",
  "client": "Įmonės pavadinimas"
}

```

13.1.2 GET struktūra

```

{
  "count": 1,
  "finances": [
    {
      "id": 35,
      "date": "2021-09-08 00:00:00",
      "client": "Įmonės pavadinimas",
      "content": "turinys",
      "debit": "20",
      "credit": "442",
      "amount": 9,
      "currency": "EUR"
    }
  ]
}

```

```

    }
  ]
}

```

13.1.3 POST struktūra

Lauko pavadinimas	Tipas	Privalomas	Reikšmė
date	DateTime	Taip	Data
client	String	Taip	Kliento pavadinimas
debit	String	Taip	Debeto sąskaita
debitSeries	String	Ne	Debeto dokumento serija
debitDocument	String	Ne	Debeto dokumento numeris
credit	String	Taip	Kredito sąskaita
creditSeries	String	Ne	Kredito dokumento serija
creditDocument	String	Ne	Kredito dokumento numeris
amount	Float	Taip	Suma
currency	String	Taip	Valiuta

```

{
  "count": 2,
  "finances": [
    {
      "date": "2021-09-08 01:00:00",
      "client": "Įmonės pavadinimas",
      "content": "1",
      "debit": "20",
      "debitSeries": "DDD",
      "debitDocument": "000001",
      "credit": "442",
      "creditSeries": "KKK",
      "creditDocument": "000002",
      "amount": 10,
      "currency": "EUR"
    },
    {
      "date": "2021-09-08 02:00:00",
      "client": "Įmonės pavadinimas",
      "content": "2",
      "debit": "20",
      "credit": "442",
      "amount": 20,
      "currency": "EUR"
    }
  ]
}

```

13.2 Klaidų pranešimai

- Error on saving to database Klaida gaunama bandant įrašyti duomenis į duomenų bazę
- Unknown Credit [%s] Tokios kredito reikšmės nėra aprašyta sąskaitų plane
- Unknown Debit [%s] Tokios kredito reikšmės nėra aprašyta sąskaitų plane
- Empty Client tag Neužpildytas kliento pavadinimas
- Unknown Client [%s] Klientas su tokiu pavadinimu nerastas. Klientą reikia sukurti atskirai prieš įterpiant įrašą.

14 Sisteminiai metodai

14.1 /system/version

Jungties versijos atvaizdavimas

```
{  
  "Name": "BAPI",  
  "Description": "Būtenta API Service",  
  "Version": "vX.X.X",  
  "Build": "",  
  "OS": "linux",  
  "Arch": "amd64",  
  "CPU": 4,  
  "GOVersion": "goX.X.X"  
}
```

14.2 /system/update

Paleidžiamas atnaujinimo tikrinimas ir atnaujinimas. Jeigu naujos versijos nėra, atnaujinimas nevyks, jeigu yra, tada per 30-60 sekundžių atsinaujins ir persikraus jungtis. Pasitikrinti galima su `version` metodu.

15 POST Rezultatas

15.1 Struktūra

Struktūra naudojama grąžinti importo rezultatams.

15.1.1 IDs

Naudojamas grąžinti sėkmingai importuotų įrašų ID ir pavadinimus pagal kuriuos buvo sukurtas įrašas.

Nr.	Lauko pavadinimas	Tipas	Privalomas	Reikšmė
1	id	Integer	Taip	Įrašo ID
2	name	String	Taip	Įrašo pavadinimas

```
{
  "id": 0,
  "name": ""
}
```

15.1.2 Error

Naudojamas grąžinti nepavykusių importuoti duomenų pranešimams

Nr.	Lauko pavadinimas	Tipas	Privalomas	Reikšmė
1	message	String	Taip	Klaidos pranešimas
2	data	Various	Taip	Grąžinami su užklausa siųstas įrašas

```
{
  "message": "",
  "data": {
    /* ... */
  }
}
```

15.1.3 Result

Naudojamas grąžinti informaciją apie sukeltus ir klaidingus įrašus. Šiuo atveju visada bus grąžinama HTTP 200 OK

Nr.	Lauko pavadinimas	Tipas	Privalomas	Reikšmė
1	imported	Integer	Taip	Sėkmingai importuotų įrašų skaičius
2	ids	[]IDs	Ne	Sėkmingai importuotų įrašų ID ir pavadinimai
3	completed	Various	Ne	Sėkmingai įkeltų duomenų sąrašas
3	warnings	Various	Ne	Įkelti bet su perspėjimais
3	errors	[]Error	Ne	Klaidų pranešimai ir jų duomenys.

```
{
  "imported": 0,
  "ids": null,
  "completed": null,
  "warnings": null,
  "errors": null
}
```

16 Kita

16.1 Pavyzdžiai

16.1.1 Sertifikato generavimas

```
openssl req -sha256 -newkey rsa:4096 -nodes -keyout key.pem -x509 -days 365 \  
-out cert.pem
```